Anti-Entropy Bandits for Geo-Replicated Consistency

Benjamin Bengfort
University of Maryland
bengfort@cs.umd.edu

Pete Keleher
University of Maryland
keleher@cs.umd.edu

ABSTRACT

Eventual consistency systems can be made more consistent by improving the visibility of a write, that is the time until a write is fully replicated. Gossip based anti-entropy methods scale well but random selection of anti-entropy partners results in less than efficient replication. We propose a simple improvement to pairwise, bilateral anti-entropy; instead of uniform random selection we introduce reinforcement learning mechanisms which assign selection probabilities to replicas most likely to have information. The result is efficient replication, faster visibility, and higher consistency, while still providing high availability and partition tolerance.

INTRODUCTION

A distributed system is made highly available when individual servers are allowed to operate independently without coordination that may be prone to failure or high latency. The independent nature of the server's behavior means that it can immediately respond to client requests, but that it does so from a limited, local perspective which may be inconsistent with another server's response. If individual servers in a system were allowed to remain wholly independent, individual requests from clients to different servers would create a lack of order or predictability, a gradual decline into inconsistency, e.g. the system would experience *entropy*. To combat the effect of entropy while still remaining highly available, servers engage in *anti-entropy sessions* [8] at a routine interval, a process that occurs in the background of client requests.

Anti-entropy sessions synchronize the state between servers ensuring that, at least briefly, the local state is consistent with a portion of the global state of the system. If all servers engage in anti-entropy sessions, the system is able to make some reasonable guarantees about the timeliness of responses; the most famous of which is that in the absence of requests the system will become consistent, eventually. More specifically, inconsistencies in the form of stale reads can be bound by likelihoods that are informed by the latency of anti-entropy sessions and the size of the system [1]. Said another way, overall consistency is improved in an eventually consistent system by decreasing the likelihood of a stale read, which is tuned by improving the *visibility latency* of a write, the speed at which a write is propagated to a significant portion of servers. This idea has led many system designers to decide that "eventual consistency is consistent enough" [2, 9] particularly in a data center context where visibility latency is far below the rate of client requests, leading to practically strong consistency.

	Pull	Push	Total
Synchronize at least 1 object	0.25	0.25	0.50
Synchronize multiple objects	0.05	0.05	0.10
Latency <= 5ms (local)	0.10	0.10	0.20
Latency <= 100ms (regional)	0.10	0.10	0.20
Total	*0.50*	*0.50*	*1.00*

Table 1: Reward Function

Recently there have been two important changes in considerations for the design of such systems that have led us to reevaluate propagation speed: systems are getting larger and are becoming geographically distributed outside of the datacenter. Scaling an eventually consistent system to dozens or even hundreds of nodes increases the radius of the network, which leads to increased noise during anti-entropy; e.g. the possibility that an anti-entropy session will be between two already synchronized nodes. Geographic distribution and extra-datacenter networks increase the latency of anti-entropy sessions so that inconsistencies become more apparent. Large, geographically distributed systems are becoming the norm – from content delivery systems that span the globe, to mobile applications, to future systems such a automated vehicular networks, and all will require additional consistency guarantees without sacrificing availability.

We propose a new class of adaptive distributed data systems whose replicas monitor their environment and modify their behavior to optimize consistency. Anti-entropy utilizes gossip and rumor spreading to efficiently propagate updates in a deterministic fashion without saturating the network [3, 4, 7]. These protocols utilize uniform random selection of peers to synchronize with, which means that a write occurring at one replica is not efficiently propagated across the network. We propose the use of *multi-armed bandit* algorithms [5, 6] to modify the probability of peer selection in order to optimize for fast, successful synchronizations. The result is a network topology that emerges according to access patterns and network latency, often localizing replicas to produce efficient synchronization, efficiency which lowers visibility latency and increases consistency.

SYSTEM DESCRIPTION

A basic sketch of an eventually consistent system is as follows:

BANDIT APPROACHES

EXPERIMENTS

DISCUSSION

CONCLUSION

REFERENCES

[1] Peter Bailis, Shivaram Venkataraman, Michael J. Franklin, Joseph M. Hellerstein, and Ion Stoica. [n. d.]. Quantifying eventual consistency with PBS. 23, 2 ([n. d.]), 279–302. http://link.springer.com/article/10.1007/s00778-013-0330-1

PaPoC'18 Worshop Proposal, April 2018, Porto, Portugal
.

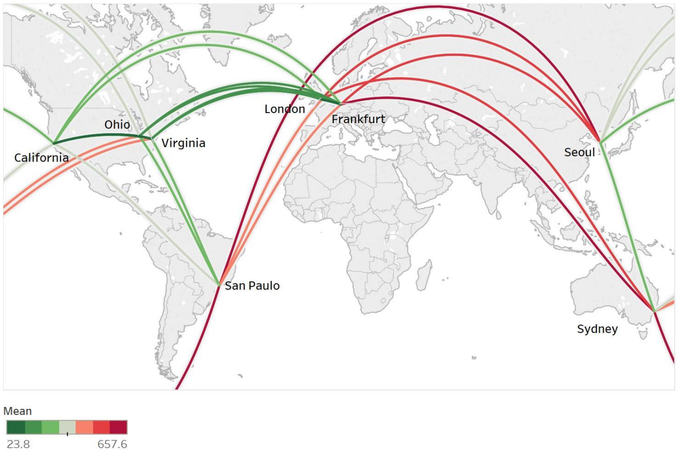

Figure 1: Geographically distributed network

[2] David Bermbach and Stefan Tai. [n. d.]. Eventual consistency: How soon is eventual? An evaluation of Amazon S3's consistency behavior. In *Proceedings of the 6th Workshop on Middleware for Service Oriented Computing* (2011). ACM, 1. http://dl.acm.org/citation.cfm?id=2093186

[3] Bernhard Haeupler. [n. d.]. Simple, fast and deterministic gossip and rumor spreading. 62, 6 ([n. d.]), 47. http://dl.acm.org/citation.cfm?id=2767126

[4] Richard Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vocking. [n. d.]. Randomized rumor spreading. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on* (2000). IEEE, 565–574. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=892324

[5] John Langford and Tong Zhang. [n. d.]. The Epoch-Greedy Algorithm for Multi-Armed Bandits with Side Information. In *Advances in Neural Information Processing Systems* (2008). 817–824. bibtex: langford_epoch-greedy_2008.

[6] Haipeng Luo, Alekh Agarwal, and John Langford. [n. d.]. Efficient Contextual Bandits in Non-stationary Worlds. ([n. d.]).

[7] Yamir Moreno, Maziar Nekovee, and Amalio F. Pacheco. [n. d.]. Dynamics of rumor spreading in complex networks. 69, 6 ([n. d.]), 066130. http://journals.aps.org/pre/abstract/10.1103/PhysRevE.69.066130

[8] Douglas B. Terry, Alan J. Demers, Karin Petersen, Mike J. Spreitzer, Marvin M. Theimer, and Brent B. Welch. [n. d.]. Session guarantees for weakly consistent replicated data. In *Parallel and Distributed Information Systems, 1994., Proceedings of the Third International Conference on* (1994). IEEE, 140–149. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=331722

[9] Hiroshi Wada, Alan Fekete, Liang Zhao, Kevin Lee, and Anna Liu. 2011. Data Consistency Properties and the Trade-offs in Commercial Cloud Storage: the Consumers' Perspective.. In *CIDR*, Vol. 11. 134–143.

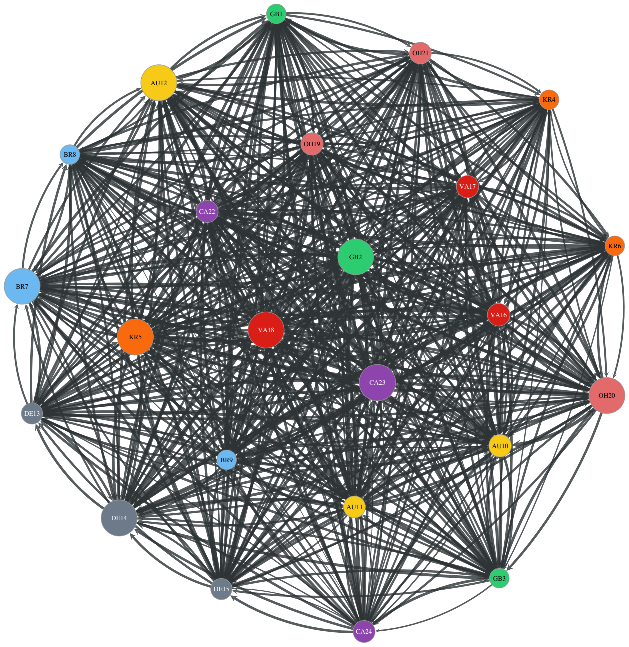

Figure 2: Uniform Selection

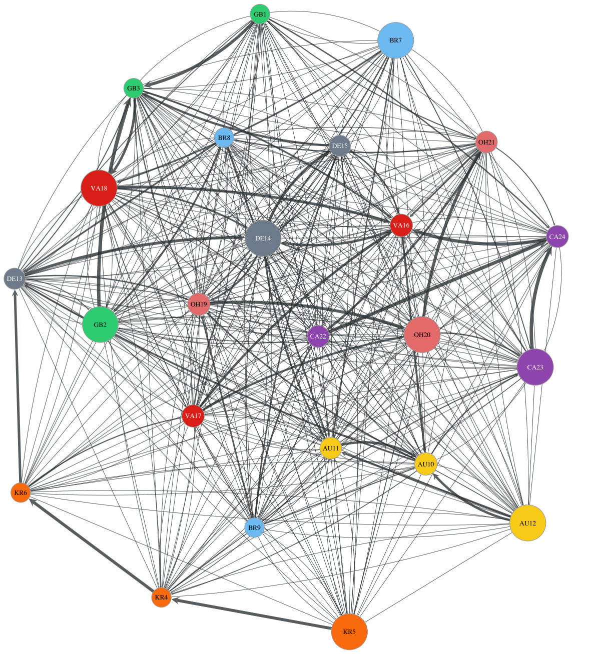

Figure 3: Epsilon Greedy $\epsilon = 0.1$

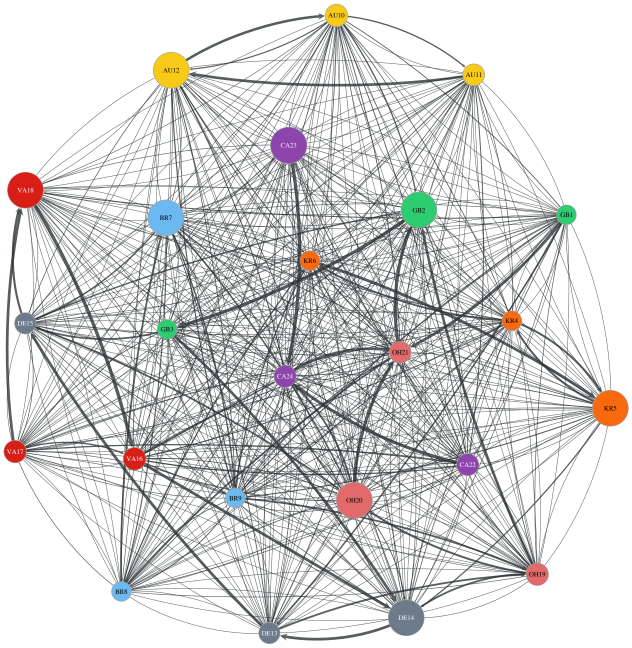

Figure 4: Epsilon Greedy $\epsilon = 0.2$

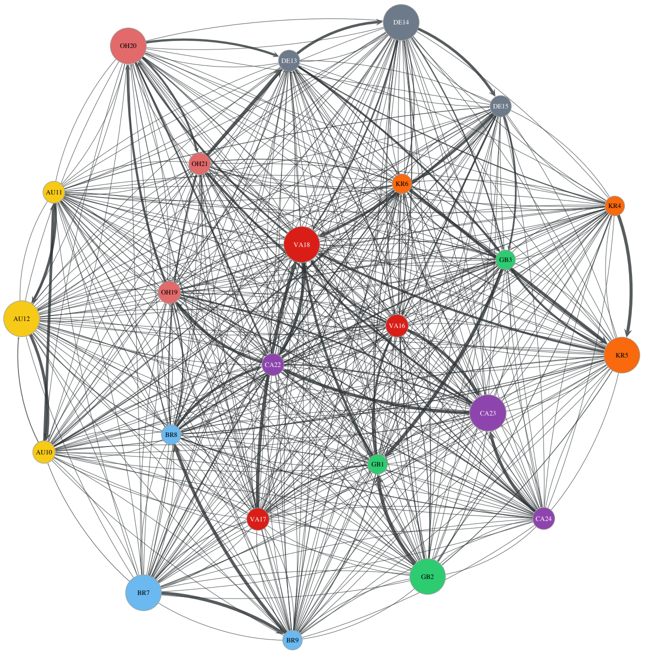

Figure 5: Annealing Epsilon

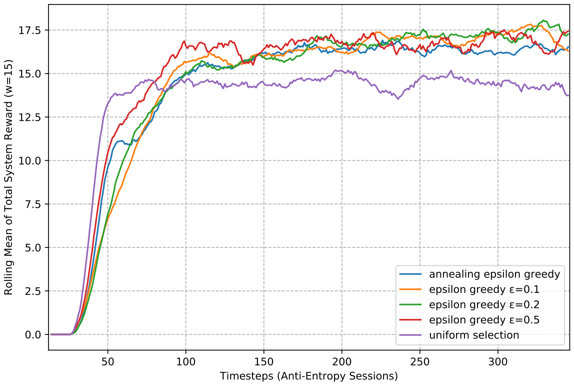

Figure 6: Total system rewards over time